Another relaxation for the multiple-choice knapsack problem

Hiroshi Iida

e-mail: *oggi@res.otaru-uc.ac.jp*

Abstract

In this paper we study a relaxation for the multiple-choice knapsack problem. In the literature the linear programming relaxation is utilized for solving the problem, however, the relaxation does not work in a particular case where the profit is equal to the weight on all items. Focusing on this issue, an attempt is made to develop another relaxation which will work especially in the hard case.

Keywords: knapsack problem; multiple-choice constraint; linear programming relaxation; subset-sum problem

1 Introduction

The Multiple-Choice Knapsack (MCK, for short) problem was proposed by Nauss [10], and the problem has been intensively studied in the last two decades. The classical 0-1 knapsack problem (KP, for short) is to pack items into a knapsack of weight limit so that the total profit of the packed items is maximal. Furthermore, on MCK, all items are split into several classes so that any pair of the classes is mutually disjoint, and we should select just one item in each class respectively. The MCK is formulated as follows:

$$\text{maximize} \quad \sum_{i \in M} \sum_{j \in N_i} p_{ij} x_{ij} \tag{1}$$

$$\text{subject to} \quad \sum_{i \in M} \sum_{j \in N_i} w_{ij} x_{ij} \leq c \tag{2}$$

$$\sum_{j \in N_i} x_{ij} = 1, \; i \in M \tag{3}$$

$$x_{ij} \in \{0,1\}, \; i \in M, \; j \in N_i, \tag{4}$$

where $M := \{1, 2, \ldots, m\}$, and the $m (\geq 2)$ indicates the number of classes. As mentioned previously, any pair of the classes N_i ($i \in M$) is mutually disjoint. Throughout this paper we call the constraint (3), which is peculiar to MCK, ***multiple-choice constraint.*** In addition the profit p_{ij} and weight w_{ij} of any j-th item in any class N_i, and the capacity c are positive integers.

Without loss of generality we will assume that

$$\sum_{i \in M} \min_{j \in N_i} w_{ij} < c, \; \sum_{i \in M} \max_{j \in N_i} w_{ij} > c, \; \text{and} \; \sum_{i \in M} \min\{w_{ij} | \max_{j \in N_i} p_{ij}\} > c$$

in order to exclude an infeasible problem, a trivial problem and a problem solvable in linear computing time for optimality, and that

$$w_{ij} + \sum_{k \in M, k \neq i} \min_{l \in N_k} w_{kl} \leq c$$

for any i, j in order to exclude an unpromising item. In addition on the cardinality of each class N_i, which we denote with $|N_i|$, we will assume that the maximal cardinality of the classes is greater than two, because MCK can be reduced to KP not in the case. Also in [10] the following theorem, which is a little modified, is presented. It is efficient to reduce MCK (see Sinha and Zoltners [12], TABLE II in p.511).

Theorem (Nauss) If $p_{ij_1} > p_{ij_2}$ and $w_{ij_1} \leq w_{ij_2}$ holds, then $x_{ij_2} = 0$ in any optimal solution to MCK.

Proof. Assuming an optimal solution in which $x_{ij_2} = 1$, we can show a contradiction to the optimality by replacing $x_{ij_1} = 0$, $x_{ij_2} = 1$ with $x_{ij_1} = 1$, $x_{ij_2} = 0$ in the solution, keeping the feasibility. ∎

In what follows, by this theorem, we will assume that there exist no two items so that each of them has the same weight in each class.

On MCK, the linear programming relaxation (also called continuous relaxation) in which the integrality constraint (4) is relaxed, that is, replaced with $0 \leq x_{ij}$ for all i, j is well-used. We call the resulting problem Linear MCK (LMCK, for short) problem. In the literature, e.g. [12][1], Armstrong et al [1][2], Dyer et al [6], Dudziński and Walukiewicz [4], and Pisinger [11][3], the LMCK is utilized for solving MCK. The following proposition is efficient to reduce LMCK.

Proposition (Sinha and Zoltners) Assume that $p_{ij_1} < p_{ij_2} < p_{ij_3}$ and $w_{ij_1} < w_{ij_2} < w_{ij_3}$. If $(p_{ij_1} - p_{ij_2})/(w_{ij_1} - w_{ij_2}) < (p_{ij_2} - p_{ij_3})/(w_{ij_2} - w_{ij_3})$ holds, then $x_{ij_2} = 0$ in any optimal solution to LMCK.

Proof. See [12].

From this if follows that the optimal solution of LMCK consists of the only item which forms an upper convex boundary in each class (see [11], Fig. 1 in p.396). Hereafter we assume for simplicity that all items are sorted in ascending order of the weight in each class respectively so that $w_{i1} < w_{i2} < \cdots$ for any $i \in M$.

On LMCK, after excluding unpromising items by Proposition, we know that finding an optimal value is equivalent to finding an optimal *slope* $s_{ij} = (p_{i,j+1} - p_{ij})/(w_{i,j+1} - w_{ij})$ satisfying $W + w_{ij} \leq c < W + w_{i,j+1}$, where $W = \Sigma_{k \neq i} \{w_{k,l+1} \mid \min_{s_{kl} \geq s_{ij}} s_{kl}\}$. If there exists no slope greater than or

[1]In p.508, the p_{q+1} in Step 3 should be replaced with $p_q + 1$.

[2]In p.189, the $p(q)$ in Step 6 should be replaced with $P(q)$, and the $1 - x_{q,P(q)} + 1$ in Step 7 should be replaced with $1 - x_{q,P(q)+1}$.

[3]In p.397, the $W + w_{ij} > c$ in Step 2 should be replaced with $W + w_{ij} - w_{i,j-1} > c$. Also, in p.401, the $\{1, \ldots, l_1\}$ in Step 5 should be replaced with $\{1, \ldots, l_i\}$.

equal to s_{ij} in some class N_k then we take w_{k1} in the class for W. Once the optimal slope is found, the optimal value is easily gained as

$$\sum_{k \neq i} \left\{ p_{k,l+1} \,\middle|\, \min_{s_{kl} \geq s_{ij}} s_{kl} \right\} + \frac{p_{ij}(W + w_{i,j+1} - c) + p_{i,j+1}(c - W - w_{ij})}{w_{i,j+1} - w_{ij}}, \tag{5}$$

where the first term is the one that w is replaced with p in W. Here we would like to add that, as pointed out in [10], the Lagrangian dual problem (see Fisher [7]) of MCK in which the constraint (2) is dualized, that is, relaxed by placing it into the objective function (1) is equivalent to LMCK. Namely the optimal Lagrange multiplier is corresponding to the optimal slope.

On KP, it is well-known that a particular condition onto the items induces a catastrophic behavior of the methods for KP (see, e.g. Chapter 4 in Martello and Toth [8]). The KP with the condition is especially called *subset-sum problem* in which there exists only one type of item to be packed into the knapsack. Here, on MCK, we consider the condition, i.e.

$$p_{ij} = w_{ij} \quad \text{for all } i \in M,\ j \in N_i. \tag{6}$$

Hereafter we call the case where the condition (6) holds on MCK *subset-sum case*. In the subset-sum case, no efficient upper bound is derived from LMCK since the optimal value of LMCK is just c, which is similar to the behavior of the Dantzig upper bound for KP (Dantzig [2]). Moreover, in such a case, no item can be fathomed by Nauss' Theorem. Therefore the reduction of MCK in order to gain a more small-sized and equivalent problem is not effective. Thus the methods hitherto proposed for MCK will no doubt show poor performance in the subset-sum case. For instance, computational experiments of applying a recent method to the MCK with (6) are presented in [11].

On the other hand, Dyer [5] solves LMCK by means of the dual problem of it. By the duality theorem, however, it should also produce an optimal value egual to a capacity in the subset-sum case. Here we would like to add that the dual problem is equivalent to the Lagrangian dual problem dualized (2) as pointed out in [4] (see each definition of $v(\overline{\mathrm{MD}})$ and $v(\mathrm{ML}(u))$). Roughly speaking, for MCK, the linear programming relaxation, the Lagrangian dual problem and the dual problem are the same each other. In Section 2 we will present a unique relaxation for MCK, and dig up its characteristics. In the final section we discuss two topics for future research.

2 Another relaxation for MCK

In this section we first present a constraint equivalent to the multiple-choice constraint (3) in the MCK (1)–(4), and second present a relaxation based on the equivalent constraint.

In the formulation of MCK, except (4), only the multiple-choice constraint is not of a form including $\Sigma_{i\in M}\Sigma_{j\in N_i}$. Then we devise a constraint of such a form:

$$\sum_{i\in M}\sum_{j\in N_i} d^{i-1}x_{ij} = r, \text{ where } d = \max_{i\in M}|N_i|, \; r = \frac{d^m-1}{d-1}. \tag{7}$$

Note that $d > 2$ as mentioned previously. The equivalency between the constraint (3) and (7) in MCK is proved as follows (we show only (7) $\Rightarrow$ (3)):

Proof. We assume that the hypothesis holds in the case where $m = k$. In the case where $m = k+1$, suppose that $\Sigma_{j\in N_{k+1}} x_{k+1,j} = 0$. Then

$$(d-1) + d(d-1) + \cdots + d^{k-1}(d-1) = d^k - 1 < d^k.$$

Therefore we should take at least one item in the class N_{k+1}.
Also suppose that $\Sigma_{j\in N_{k+1}} x_{k+1,j} = 2$. Then, on $N_1, N_2, \ldots, N_{k+1}$, the sum of d^{i-1} for $1 \le i \le k+1$ is

$$1 + d + d^2 + \cdots + d^k = \frac{d^k-1}{d-1} + d^k < 2d^k - 1 < 2d^k.$$

Hence we cannot take two more items in N_{k+1}. ∎

By this, we have another formulation of MCK as follows:

$$\begin{aligned} &\text{maximize} && \sum_{i\in M}\sum_{j\in N_i} p_{ij}x_{ij} \\ &\text{subject to} && \sum_{i\in M}\sum_{j\in N_i} w_{ij}x_{ij} \le c \\ &&& \sum_{i\in M}\sum_{j\in N_i} d^{i-1}x_{ij} = r,\ d = \max_{i\in M}|N_i|,\ r = \frac{d^m-1}{d-1} \\ &&& x_{ij} \in \{0,1\},\ i \in M,\ j \in N_i. \end{aligned} \tag{8}$$

Now we present a relaxation for MCK by (2)+(7):

$$\begin{aligned} &\text{maximize} && \sum_{i\in M}\sum_{j\in N_i} p_{ij}x_{ij} \\ &\text{subject to} && \sum_{i\in M}\sum_{j\in N_i} (w_{ij}+d^{i-1})x_{ij} \le c + r \\ &&& \sum_{i\in M}\sum_{j\in N_i} x_{ij} = m \\ &&& x_{ij} \in \{0,1\},\ i \in M,\ j \in N_i, \end{aligned} \tag{9}$$

where the d and r are the same as those in (8) respectively. This problem is a KP with an additional constraint on the cardinality of an optimal solution. Although the additional constraint has been hidden behind (7) so far, it should be made explicit, the reason for which will be showed afterward.

In the subset-sum case, the problem (9) will be a hard instance of KP in which the profits and weights are strongly correlated. Moreover the problem (9) is not tractable in recent computer systems even if a 64-bit processor, because the numerical data appeared in (9) are too large, especially the d^m is. The size of a problem tractable by our relaxation will be at most $d = m = 10$ in a 32-bit processor. Still, in the subset-sum case, the problem (9) seems to be better than LMCK since the profit-to-weight ratio of the items is not constant. In the following we provide several tiny subset-sum examples ($m=2$, $d=3$) to observe how our relaxation behaves.

Example 1. Let $w_{11} = 2$, $w_{12} = 5$, $w_{13} = 6$ in N_1, $w_{21} = 3$, $w_{22} = 5$, $w_{23} = 8$ in N_2, and $c = 12$.

In this example the capacity is $c + 1 + d = 16$. Then an optimal value 11 less than c is favorably gained by $x_{12} = x_{13} = 1$ and other x_{ij}'s are zero (also gained by $x_{13} = x_{22} = 1$ and other x_{ij}'s are zero). We remark that if we exclude the constraint on the cardinality of an optimal solution, an optimal value 13 is obtained by $x_{11} = x_{12} = x_{13} = 1$.

Here is an additional remark that, in the case where the two classes are exhanged, an optimal value 13 is obtained by $x_{12} = x_{13} = 1$. Our relaxation thus depends on the order of classes. Moreover this implies that our relaxation will produce a problem of an optimal value greater than a capacity in some cases.

Example 2. Let $w_{11} = 5$, $w_{12} = 10$, $w_{13} = 18$ in N_1, $w_{21} = 4$, $w_{22} = 11$, $w_{23} = 16$ in N_2, and $c = 24$.

The capacity is 28, and an optimal value 23 is gained by $x_{11} = x_{13} = 1$. In the case where the two classes are exchanged, an optimal value 22 is gained by $x_{11} = x_{23} = 1$.

Example 3. Let $w_{11} = 1$, $w_{12} = 4$, $w_{13} = 5$ in N_1, $w_{21} = 1$, $w_{22} = 3$, $w_{23} = 6$ in N_2, and $c = 7$.

The capacity is 11, and an optimal value 9 is obtained by $x_{12} = x_{13} = 1$. In the case where the two classes are exchanged, the same optimal value 9 is obtained by $x_{12} = x_{13} = 1$.

As implied by Example 3, the problem (9) will not work in the case where both the m and d are even large, because we can select items of fairly large weight from N_i's of small i without exceeding the enlarged capacity. Thus our relaxation is not practical in general, however, at least it shows that there exists a relaxation which will produce a problem of an optimal value not equal to a capacity against the subset-sum case, in contrast to the linear programming relaxation.

3 Notes

In this section we would like to add two topics: First, we present a constraint also equivalent to the multiple-choice constraint, which is formulated as follows:

$$\sum_{i\in M}\sum_{j\in N_i}\frac{x_{ij}}{d^{i-1}}=r',\ \text{where}\ d=\max_{i\in M}|N_i|,\ r'=\frac{1}{d^{m-1}}\ \frac{d^m-1}{d-1}.$$

With this constraint we can consider a relaxation for MCK by the same way as the one for (7), however, the resulting problem will have the same defects as (9). Moreover the problem will have still another defect in exchange for the intractability, i.e. the accuracy of computing.

Second, we can also consider Lagrangian relaxation for the MCK (8) in which the constraint (7) is dualized:

$$\begin{aligned} z(\lambda) \ &= \ \max_x \sum_{i\in M}\sum_{j\in N_i}(p_{ij}-\lambda d^{i-1})x_{ij}+\lambda r \\ \text{subject to}\quad & \sum_{i\in M}\sum_{j\in N_i} w_{ij}x_{ij}\le c \\ & x_{ij}\in\{0,1\},\ i\in M,\ j\in N_i, \end{aligned} \tag{10}$$

where λ is a Lagrange multiplier without requirements. This problem is a KP, however, it will tend to be a hard instance especially in the subset-sum case. Recently an efficient algorithm for a hard KP was developed by Martello and Toth [9], and it will be promising to solve (10) when $|\lambda|\ll 1$.

On the problem (10) how we find an appropriate λ so that it makes $z(\lambda)$ small, which is an issue we should address. Note that it should be best to find a λ which minimizes $z(\lambda)$ if possible, whereas the resulting problem is not tractable in the recent computer systems in the case where $|\lambda|$ is large. From a practical point of view, it is preferable that $|\lambda|\ll 1$. Also, in the event that the problem (10) with an appropriate λ does not give a tight upper bound for MCK, the constraint on the cardinality of an optimal solution, which is appeared in (9), would be helpful to improve (10).

References

[1] Armstrong, R.D., D.S. Kung, P. Sinha and A.A. Zoltners, "A Computational Study of a Multiple-Choice Knapsack Algorithm," *ACM Trans. on Math. Software* **9**(2) 184–198 (1983).

[2] Dantzig, G.B., "Discrete variable extremum problems," *Operations Res.* **5**, 266–277 (1957).

[3] Dudziński, K. and S. Walukiewicz, "A fast algorithm for the linear multiple-choice knapsack problem," *Oper. Res. Lett.* **3**(4) 205–209 (1984).

[4] Dudziński, K. and S. Walukiewicz, "Exact methods for the knapsack problem and its generalizations," *Euro. J. Opnl. Res.* **28**, 3–21 (1987).

[5] Dyer, M.E., "An $O(n)$ algorithm for the multiple-choice knapsack linear problem," *Math. Programming* **29**, 57–63 (1984).

[6] Dyer, M.E., N. Kayal and J. Walker, "A branch and bound algorithm for solving the multiple-choice knapsack problem," *J. of Computational and Appl. Math.* **11**, 231–249 (1984).

[7] Fisher, M.L., "The Lagrangian Relaxation Method for Solving Integer Programming Problems," *Management Sci.* **27**(1) 1–18 (1981).

[8] Martello, S. and P. Toth, *Knapsack Problems: Algorithms and Computer Implementations,* John Wiley & Sons, Chichester, England, 1990.

[9] Martello, S. and P. Toth, "Upper bounds and algorithms for hard 0-1 knapsack problems," *Operations Res.* **45**(5) 768–778 (1997).

[10] Nauss, R.M., "The 0-1 knapsack problem with multiple choice constraints," *Euro. J. Opnl. Res.* **2**, 125–131 (1978).

[11] Pisinger, D., "A minimal algorithm for the Multiple-Choice Knapsack Problem," *Euro. J. Opnl. Res.* **83**, 394–410 (1995).

[12] Sinha, P. and A.A. Zoltners, "The Multiple-Choice Knapsack Problem," *Operations Res.* **27**(3) 503–515 (1979).

[13] Zemel, E., "The Linear Multiple Choice Knapsack Problem," *Operations Res.* **28**(6) 1412-1423 (1980).

[14] Zemel, E., "An O(n) algorithm for the linear multiple choice knapsack problem and related problems," *Information Processing Letters* **18**(3) 123-128 (1984).